

A Simulator to Assess Energy-Saving Techniques in Content Distribution Networks*

Tom Bostoen¹, Jeff Napper¹, Sape Mullender¹, and Yolande Berbers²

¹ Alcatel-Lucent Bell Labs,
Copernicuslaan 50, B-2018 Antwerpen, Belgium,
{Tom.Bostoen, Jeff.Napper, Sape.Mullender}@alcatel-lucent.com,
<http://www.alcatel-lucent.com/bell-labs>
² Katholieke Universiteit Leuven,
Celestijnenlaan 200A, B-3001 Heverlee, Belgium,
Yolande.Berbers@cs.kuleuven.be,
<https://distrinet.cs.kuleuven.be>

Abstract. The scalable and bandwidth-efficient delivery of IPTV services to an increasingly diverse set of screens requires the deployment of *telco* content distribution networks (CDNs). These CDNs are composed of cache servers located in the telco's data centers close to the end user. The additional cache servers need to be designed for energy efficiency to limit the increase of data-center energy consumption. We analyze real HTTP adaptive-streaming traces from an operational telco CDN delivering IPTV to mobile devices to identify workload characteristics that can be exploited to conserve energy. We also present a trace-driven simulator that models the energy consumption of such a CDN down to the level of the cache-server disk to evaluate potential energy-saving techniques. The traces reveal cyclic load fluctuations that can be exploited to save energy in CDNs by varying the number of powered cache servers and disks according to the load.

Keywords: Content distribution network, HTTP adaptive streaming, cache server, energy efficiency, disk drive, power reduction

1 Introduction

Because consumers want to watch videos on every screen they own, Internet service providers are starting to extend their IPTV offering with online multi-screen video services. ISPs can deliver video streams from content providers to their clients by means of their own content distribution network (CDN) for a superior viewing experience. Such *telco* CDNs [1], which are composed of disk-packed cache servers deployed in the ISP-owned regional network close to the

* ©Springer-Verlag Berlin Heidelberg, 2014. This is the authors' version of the original paper published in the Proceedings of the 2nd International Workshop on Energy-Efficient Data Centers (E²DC '13), Springer LNCS 8343, pp. 83-98.

end users, ensure a scalable, bandwidth-efficient delivery of both linear and on-demand video streams. Data centers consume globally 1 to 2% of all available power and their power consumption grows by 15% a year [2]. The deployment of additional power-hungry cache servers in data centers to build new telco CDNs can be expected to accelerate the increase in data-center power consumption.

In this paper, we target energy savings in content distribution networks by approaching the CDN as a storage system distributed over multiple data centers, thereby exploiting the state-of-the-art in power-aware storage systems. We present the following three research contributions. As our main contribution, we propose a trace-driven CDN energy-consumption simulator based on the common linear model for the energy consumption of a cache server as a function of the server load [3] and the known model for the energy consumption of a disk drive as a function of disk read and write accesses [4]. The simulator includes a model for traditional hard disk drives as well as solid-state disks. It allows transitioning a cache server from the idle mode to the standby mode and the other way around. Such transitions can also be applied to individual disks of a cache server. The simulator allows trading off energy consumed by the caches (and their disks) against CDN scalability (or throughput) and bandwidth efficiency. For this initial release of the simulator, we don't model the location of the caches in the network. Consequently, the load balancing over the caches doesn't take the proximity between client and cache into account. We intend to support geographical load balancing in the next iteration of the simulator. We plan to release the simulator as free and open-source software to accelerate the research in this domain through collaboration.

As a supporting second contribution, we provide a first-cut characterization of the multiscreen (tablet and smartphone) IPTV workload used to drive the simulator. We present the most relevant workload characteristics of the HTTP adaptive streaming traces collected from an operational telco CDN delivering exclusively IPTV services. The file download bandwidth as a function of time exhibits a diurnal pattern of large load fluctuations to a great extent consistent with observations for similar workloads, e.g. IPTV delivered to the traditional TV screen only [5] and user-generated video-on-demand [6].

The simulator is based on known models. Therefore, we don't present a validation of the simulator. Instead, we present interesting simulation results based on the IPTV workload traces at the level of the disk, cache server, and CDN as derived third contribution. Simulation reveals that CDNs are far from *energy-proportional* [3]. This lack of energy proportionality leads to energy waste because of the fluctuations in the workload. This waste can be reduced by applying a power-reduction technique known for data-center storage systems as DIV [7] (short for diverted accesses) to content distribution networks.

The remainder of the paper has the following outline. Section 2 describes the workload characterization. In Section 3, we present the CDN simulator. Simulation results are described in Section 4. In Section 5, we describe related work. Finally, Section 6 is the conclusion of our paper.

2 Workload Characterization

In this section, we search for CDN workload characteristics that can be exploited to save energy in the CDN. We start from workload traces produced by an operational telco CDN delivering IPTV (live broadcast TV and video-on-demand) to mobile screens (smartphone and tablet) by means of HTTP adaptive streaming. The combination of the widespread use of wireless terminals, even in the home, and the large bandwidth variation exhibited by wireless connections drives the adoption of HTTP adaptive streaming as the protocol for delivering on-line video services [8]. For HTTP adaptive streaming, the source video stream is encoded at different bit rates. These different-bit-rate streams are partitioned in short, constant-duration segments. The segment files are stored in the origin server together with index files, which provide an ordered list of references to segment files. For live television, the index files provide a frequently regenerated rolling snapshot of the broadcast. Using HTTP, the client fetches segments one after the other (according to the index) from a cache, which might then fetch it from the origin. The client selects every segment such that the segment's bit-rate encoding is adapted to the network conditions.

The traces contain information about all HTTP responses delivered by any of the CDN caches during one calendar day. We analyze the seven consecutive traces of a week during the Fall of 2012. We subsampled the traces by randomly selecting clients. The resulting subsampled traces contain the information of about ~ 1.4 billion HTTP responses and have a total zipped-file size of ~ 48 GiB. Our analysis results in (1) the probability mass function (pmf) of the downloaded file and (2) the file download bandwidth as a function of time. Figure 1 shows the estimated pmf of the downloaded file. We consider all files (solid line), the index files only (dashed line), and the segment files only (dash-dot line). The pmf estimation involves counting the number of downloads per unique file during one week. This table of download frequencies is divided by the total number of unique files. This division leads to relative frequencies. The files are ranked according to their download frequency. The roughly ~ 3000 files that are downloaded most frequently appear to be downloaded at a significantly higher rate than the rest of the files. These turn out to be the index files that are continuously fetched for live streams. Even when we only consider segment files, the pmf doesn't appear to be linear in the log-log scale.

Figure 2 shows the file download bandwidth as a function of time. This download bandwidth is calculated by adding up the sizes of the HTTP responses per 60-s time interval (during which the HTTP response was sent) and dividing by 60 s. The figure reveals a cyclic load fluctuation that can be explained based on end user behavior. During the day, most people are working and therefore the load on the caches is low. In the evening more people start watching videos until $\sim 21:30$ when the load on the caches peaks. After this peak, the load drops as people presumably go to sleep. In the morning ($\sim 7:30$) before people generally go to work, there is a smaller load peak. During the weekend, the load fluctuation is naturally somewhat different from the fluctuation on a weekday. In addition to the daily and weekly load variation, there might also be seasonal load fluctuations

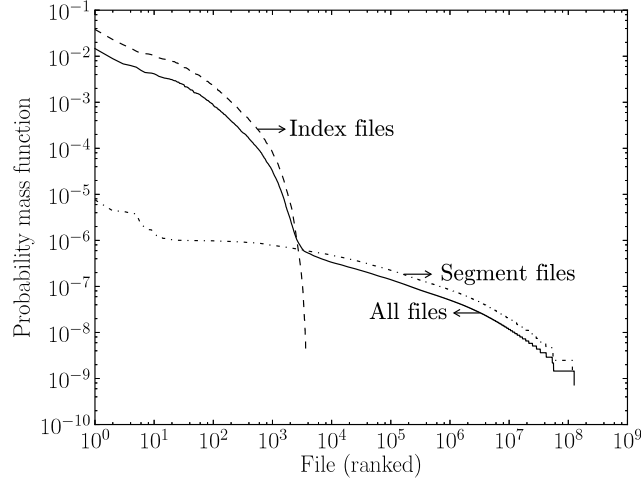


Fig. 1: Probability mass function of the (ranked) downloaded file for all files, the index files only, and the segment files only

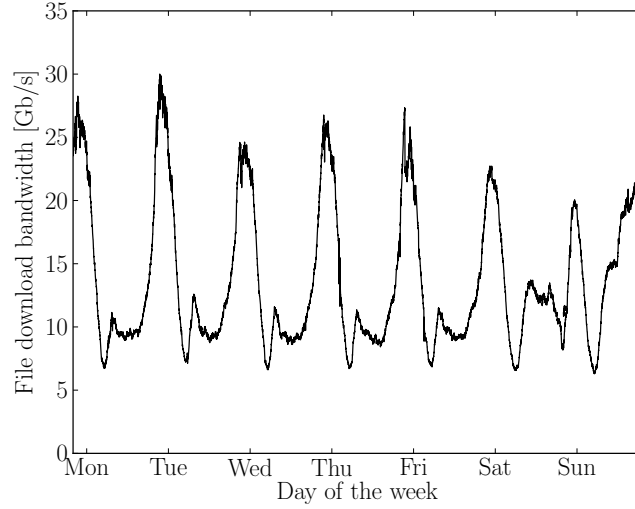


Fig. 2: File download bandwidth (from caches to clients) over a single week

as well as a gradual load variation caused by the changing number of subscribers to the service. A large load variation over long time intervals can be exploited to save energy by activating at any point in time only the number of caches and disks required to support the load at that moment, as will be explained in Section 4.

To the best of our knowledge, we are not aware of any work characterizing exactly the type of workload we consider in this paper: IPTV (live and on-demand) delivered to mobile devices (smartphones and tablets) by means of HTTP adaptive streaming over a telco CDN. Nevertheless, it does make sense to compare our results with the characteristics of slightly different workload types such as traditional IPTV, i.e., live TV delivered to TV screens, on the one hand [5] and user-generated videos streamed on-demand on the other hand [6]. We are especially interested in comparing the load fluctuations. A traditional IPTV workload shows cyclic load fluctuations similar to the ones shown in Figure 2 although we observe a slightly more pronounced peak in the morning and the large peak at $\sim 15:00$ observed in [5] is missing in Figure 2. Also a web-based video-sharing workload [6] shows daily load variations but the traffic peaks in the afternoon (between $\sim 14:00$ and $\sim 18:00$), thus earlier than IPTV.

3 CDN Energy Simulator

In this section, we describe a trace-driven CDN energy-consumption simulator, which can be used to evaluate potential techniques for saving energy in content distribution networks. Our Python-based simulator uses the workload traces described in Section 2 as input. The CDN model used as basis for our simulator is described next.

3.1 CDN Model

In practice, when a client wants to access one of the video streams on the origin, a combination of DNS indirection and HTTP redirection redirects the client to the best cache server. Commonly used request-routing policies would select the cache server that is closest to the client and not overloaded. However, our traces do not group clients geographically, and the initial release of our simulator described in this paper does not consider location. Therefore, we selected a request-routing policy for the simulator that balances the load across the cache servers without consideration of distance between client and cache. A new client to the CDN is redirected to the active cache server with the lowest load. The simulator allows turning off caches to save energy. Every succeeding HTTP request from a client gets redirected to the same cache server. Only when that cache is overloaded or powered-down does the simulated request-routing system redirect the client to another cache server, again with the lowest load.

The simulator models *cooperative pull-based* content outsourcing where cache servers don't prefetch files from the origin: a file is pulled from the origin upon a cache miss. A cache may be considered a surrogate server or reverse proxy because it distributes content on behalf of the origin. Caches located in the same rack could cooperate; for example, using the Internet Cache Protocol (ICP). When a cache doesn't have a requested file in memory or disk, it could first try to fetch this file efficiently from one of its neighboring caches in the same rack before going to the origin. Although the simulator supports cooperative content

outsourcing, we assume for simplicity that there is only one cache server per rack.

The simulated cache server applies *multi-level* caching wherein the server fetches the requested file from, in order of priority, its memory, one of its disks, one of its neighboring caches (in the same rack), or the origin. Initially in pull-based caching, a file requested by a client is not available in the cache server. The server first fetches the file from the origin (or possibly from its neighbors), then delivers it to the client, and finally caches it in its primary memory. New files are always cached in memory. When memory is full, least-recently-used (LRU) files are moved to a disk to make space available for caching a new file. The least-loaded disk is selected to balance the load across all disks. When a disk is full while a new file needs to be stored on it, least-recently-used files are removed until the new file fits in the freed space. Thus, the simulator applies the common LRU cache replacement policy both for memory and disks.

The simulator models the CDN’s energy consumption attributed to its cache servers. The energy consumption of the disks (§ 3.2) in the caches is modeled separately from the servers (§ 3.3). The simulator includes an energy-consumption model for both traditional hard disk drives as well as solid-state disks. The consumed energy is logged per time interval $T_{res} = 60$ s (by default) for the duration of the workload trace. The total energy consumed by the CDN corresponds to the total amount of energy consumed by the cache servers the CDN is composed of, and the total energy consumed by a cache server is the sum of the energy consumed by the server excluding disks and the energy consumed by each of the server’s disks. We only consider energy consumed directly by the cache servers; we ignore the additional energy required for hosting these servers in a data center (such as energy for cooling and network access). The goal of energy-saving techniques for content distribution networks is to reduce the CDN’s energy consumption without unacceptable performance degradation.

3.2 Disk Energy-Consumption Model

The power consumed by a hard disk drive (HDD) depends on its activity. When a disk is idle, it consumes P_{id}^{disk} . When seeking, the disk consumes P_{sk}^{disk} . When reading or writing, the disk is in the active power state and consumes P_{act}^{disk} . Finally, the disk can be spun down to the standby power state in which it consumes P_{sb}^{disk} . The current simulator ignores the energy required to spin a disk down or up.

When the disk receives a request i to read or write a file at time t_i , it takes (1) time T^{sk} (on average) to move the actuator arm to the required cylinder, (2) time T^{rt} (on average) to wait until the right sector has rotated under the read/write head, and (3) time T_i^{tf} to transfer the file. For sake of simplicity, we assume that every file is contiguously laid out on disk. We ignore the underestimation of the seek energy that may occur as a consequence of this assumption. The transfer time (milliseconds) is a function of the file size S_i^{file} (KiB) and the transfer rate R_{tf} (MiB/s): $T_i^{tf} = 1000S_i^{file}/(1024R_{tf})$. The total time $T_{req,i}^{disk}$ spent by

a disk to handle request i is the sum of the seek time, rotational latency, and transfer time for that request, i.e., $T_{req,i}^{disk} = T^{sk} + T^{rt} + T_i^{tf}$. The total time T_j^{disk} a disk is busy addressing all requests received during time interval j is given by $T_j^{disk} = \sum_{jT_{res} \leq t_i < (j+1)T_{res}} T_{req,i}^{disk}$. The simulator enforces the constraint $T_j^{disk} \leq T_{res}$ to avoid overloading a disk. Any request that does not fit within this constraint is addressed by the cache server by fetching the requested file directly from the origin. The total energy $E_{req,i}^{disk}$ consumed by a disk to handle request i is given by $E_{req,i}^{disk} = P_{sk}^{disk}T^{sk} + P_{id}^{disk}T^{rt} + P_{act}^{disk}T_i^{tf}$. The total energy E_j^{disk} an active disk consumes during a time interval j may be expressed as $E_j^{disk} = (T_{res} - T_j^{disk})P_{id}^{disk} + \sum_{jT_{res} \leq t_i < (j+1)T_{res}} E_{req,i}^{disk}$. A disk in the standby mode during time interval j consumes the amount of energy $E_j^{disk} = P_{sb}^{disk}T_{res}$.

Although traditional hard disk drives are usually preferred when capacity matters most, solid-state disks (SSDs) can be much faster for high I/O workloads. We model the energy consumption of solid state disks similarly as hard disk drives but with the following differences. The transfer rate for reading is different from the rate for writing. In addition, also the active read power is different from the active write power. Finally, we note that an SSD doesn't require seeking, or incur rotational latency. Our simulator supports solid-state disks and provides numeric values for all model parameters for two SSD types: HP 120GB 3G SATA SFF MDL SSD and HP 200GB 3G SATA MLC SFF [9]. These SSD types are used in cache servers deployed in operational CDNs. For the HP 120GB disk, the idle power is 0.1 W, the read power 1.5 W, the write power 1.9 W, the transfer rate for reading 230 MiB/s, and the transfer rate for writing 180 MiB/s. For the HP 200GB disk, the idle power is 1.3 W, the read power 1.8 W, the write power 3.4 W, the transfer rate for reading 257 MiB/s, and the transfer rate for writing 235 MiB/s. We assume that 80% of the disk capacity can be used for caching files. We assume that the SSD standby power is 0 W.

3.3 Cache Server Energy-Consumption Model

The power consumed by a cache server depends on the server's load. This load is determined by the server's download rate $R^{srv,dn}$, where we define downstream as the direction towards the clients. This download rate varies over time and therefore depends on the time interval j , i.e., $R_j^{srv,dn}$. The simulator calculates $R_j^{srv,dn}$ as follows: Every HTTP response i with size S_i^{resp} (in bytes) is sent by the server towards the client at time t_i during a certain time interval j (where $jT_{res} \leq t_i < (j+1)T_{res}$) and therefore adds a rate $r_i^{srv,dn} = 8S_i^{resp}/(10^9T_{res})$ (in Gb/s) to the server's download rate $R_j^{srv,dn}$ (in Gb/s), i.e., $R_j^{srv,dn} = \sum_{jT_{res} \leq t_i < (j+1)T_{res}} r_i^{srv,dn}$. The request-routing system enforces the constraint $R_j^{srv,dn} \leq R_{max}^{srv,dn}$ to avoid overloading a cache server (with maximum download rate $R_{max}^{srv,dn}$). When a client request cannot be handled by a cache server because of this constraint, the client is redirected to another cache server (the one with the lowest load). If all cache servers are overloaded, the request is dropped.

The load λ_j on the cache server during time interval j is defined by $\lambda_j = R_j^{srv,dn}/R_{max}^{srv,dn}$. This implies: $0 \leq \lambda_j \leq 1$. The power consumed by a fully-loaded ($\lambda = 1$) cache server is given by P_{max}^{srv} . However, this power includes the power consumed by the cache disks for which we use a separate disk-specific energy-consumption model (§ 3.2). We can derive the power P_{max}^{srv-} consumed by a fully-loaded cache server excluding disks by subtracting from P_{max}^{srv} the power consumed by the cache disks, i.e., $P_{max}^{srv-} \approx P_{max}^{srv} - N_{disk}P_{act}^{disk}$. The power consumed by the disks of a fully-loaded cache server is approximately the power consumed by all N_{disk} disks in the active mode P_{act}^{disk} where we assume that all disks are of the same type. For an SSD, we assume 80 % read activity and 20 % of write activity, i.e., $P_{act}^{disk} = 0.8P_r^{disk} + 0.2P_w^{disk}$.

The power consumed by an idle ($\lambda = 0$) cache server is given by P_{id}^{srv} . This power includes the power of the idle cache disks. We derive the power P_{id}^{srv-} consumed by an idle cache server excluding disks as expressed by $P_{id}^{srv-} = P_{id}^{srv} - N_{disk}P_{id}^{disk}$. The power P_j^{srv-} of a cache server excluding disks with load λ_j at time interval j is modeled as a linear function of this load, i.e., $P_j^{srv-}(\lambda_j) = P_{id}^{srv-} + \lambda_j(P_{max}^{srv-} - P_{id}^{srv-})$. The simulator allows transitioning cache servers to a standby mode in which the server consumes P_{sb}^{srv-} . Similar as for the disks, the current simulator ignores the energy required for such transition. The energy consumed by a cache server excluding disks is given by $E_j^{srv-} = P_j^{srv-}T_{res}$. Combining the results of this section and the previous one, the energy consumed by a cache server including disks is $E_j^{srv} = E_j^{srv-} + \sum_{k=0,1,\dots,N_{disk}} E_{j,k}^{disk}$.

The simulator implements this cache-server energy-consumption model and provides numeric values for all model parameters for an HP Proliant server as an example. Such a server is equipped with a Dual Intel Xeon 5600 processor, 144 GiB of DDR3 RAM, 2 300 GiB SAS disks, and 14 120 GiB SSD disks. This cache server contains $N_{disk} = 14$ cache disks of type HP 120GB specified in Section 3.2. However, we can simulate the energy consumption of such a server with any number of cache disks of any type as explained previously in this section. The server's idle energy excluding disks is 224.6 W. The server's maximum download rate is 18 Gb/s. At this rate, the server excluding disks consumes the maximum power 405.88 W. We assume the cache-server standby power to be 0 W. In addition, we assume that 80 % of the memory can be used for caching files.

4 Simulation Results

In this section, we demonstrate the capabilities of the CDN simulator and based on the simulation results, we identify DIV as a power-reduction technique applicable to content distribution networks. We present the results of a simulation of a CDN composed of two cache servers filled to capacity with disks. The cache servers are located in different racks so there is no inter-cache traffic. We drive the simulation by means of a one-day subsampled trace (as previously described) recorded on a Sunday by an operational CDN.

The simulated CDN is different from the operational CDN. For the simulation, we use the minimum number of cache servers required to serve all requests in the trace at any time interval, and we provision the maximum number of cache disks in the cache servers so as to maximize the cache hit ratio. Both cache servers in our simulation are HP Proliant servers, which are introduced in Section 3.3. The simulated servers each contain 14 solid-state cache disks of type HP 200GB as specified in Section 3.2. Since we currently don't consider location, adding more cache servers doesn't improve performance. On the contrary, if we add a cache server, the clients get distributed over more cache servers due to load balancing. This distribution leads to diminished file sharing among clients and therefore a smaller cache hit ratio. Moreover, limiting the number of cache servers to the minimum required for serving all requests also leads to a conservative estimate of the energy savings. At the start of the simulation the caches are empty. The cache memory fills up in ~ 10 min; the cache disks are full after ~ 10 h. In the following sections, we present the simulation results for (1) one of the disks, (2) one of the cache servers, and (3) the complete CDN. Note that the results are very similar for all disks and all cache servers because of the load balancing.

4.1 Disks

Per time interval, the disk's energy consumption depends on the disk reads and writes during the interval. Therefore, we first present Figure 3, which shows the disk read and write throughput as a function of time. The disks are not heavily loaded because the load is distributed over all 14 disks. The maximum observed write throughput represents roughly $\sim 10\%$ of the maximum write throughput supported by the disk (235 MiB/s), while the maximum observed read throughput corresponds to approximately only $\sim 1\%$ of the maximum read throughput that the disk can handle (257 MiB/s). Such a write-dominated workload is abnormal for cache disks and reveals an opportunity for improving the design of the simulated CDN. Provided the memory-based cache is large enough, there is no benefit to writing video segments of live streams to disk. Eliminating these unnecessary disk writes would save energy and improve the disk cache hit rate.

Because the load on the disk is small, the power consumed by the disk only exceeds the idle power at most by roughly 10% as shown in Figure 4. For this HP 200GB SSD the idle power is 1.3 W, the active read power is 1.8 W, and the active write power is 3.4 W. The load on the disk varies over time by a factor of ~ 2 , whereas the power consumed by the disk only varies by roughly 10% . This discrepancy exposes the disk's lack of energy proportionality.

4.2 Cache Servers

The energy consumed by a cache server depends on its load. The download rate determines the cache load. Figure 5 shows the download rate over a single day and divides this rate in memory cache hits, disk cache hits, and cache misses. Every cache miss requires a file upload from the origin server. The maximum

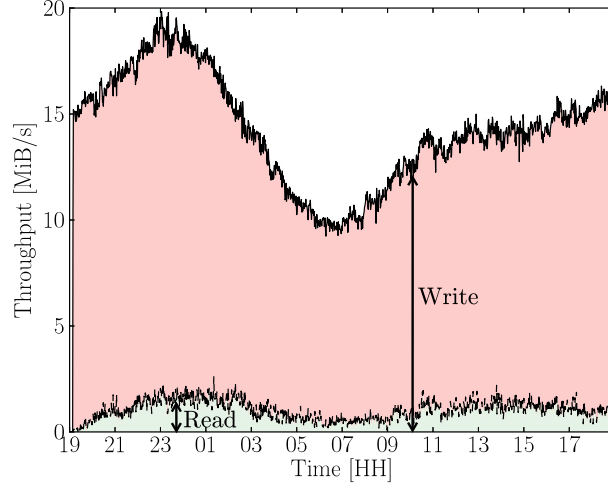


Fig. 3: Single cache-server disk read and write throughput over a single day

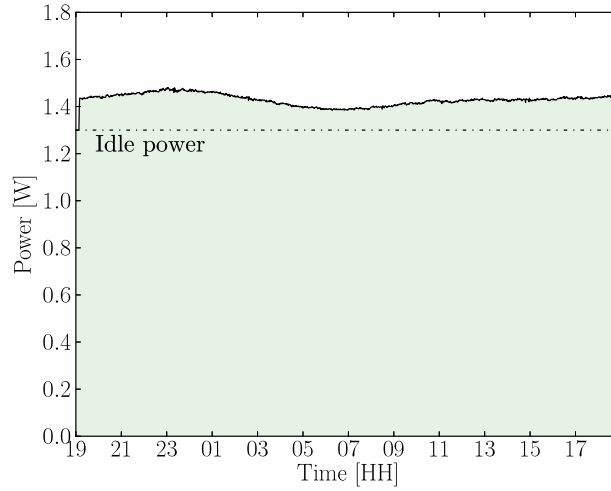


Fig. 4: Single cache-server disk power consumption

download rate for this type of HP Proliant server is 18 Gb/s. The cache hit ratio (including memory and disk cache) is approximately 80% and remains more or less constant during the entire day. There are relatively few disk cache hits because (1) the linear video streams can be served directly from primary memory because it is large enough and (2) on-demand video represents only about 5% of the data requested by clients.

The load fluctuation over the day causes a similarly fluctuating power consumption as shown in Figure 6. This figure also divides the consumed power

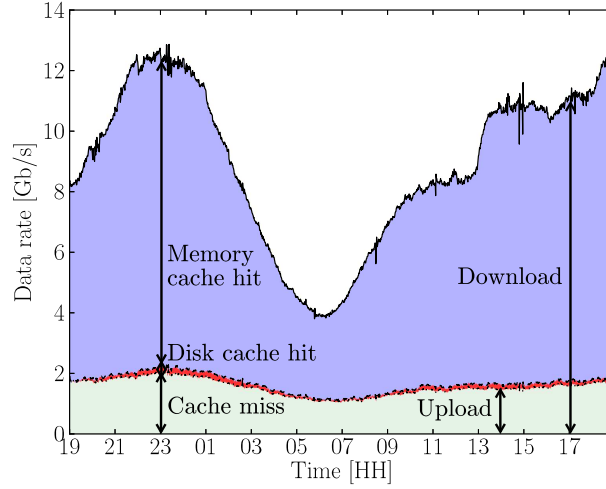


Fig. 5: Single cache-server data rate over a single day divided in memory cache hits, disk cache hits, and cache misses

between the cache disks and the cache server excluding disks. The maximum power consumption of this type of cache server (including 14 HP 200GB disks) is approximately 435.56 W, whereas the idle power consumption is 242.8 W. For the type of cache server and disks under consideration, the power consumed by the disks represents about 5 % of the total power consumed. This ratio strongly depends on the disk type and is significantly higher for hard disk drives for which we don't present simulation results in this paper to save space. Like a disk, a cache server is far from energy proportional as a comparison of Figure 6 and Figure 5 reveals.

4.3 CDN

Finally, we present simulation results at the level of the CDN by aggregating the results of the 2 cache servers and their 28 disks (in total). For comparison, we add the simulation results for a similar CDN composed of just 1 HP Proliant cache server also packed with 14 SSDs of type HP 200GB. Figure 7 shows the download and upload rate for the original 2-cache CDN and the newly-introduced 1-cache CDN over a single day. For both CDNs, the download rate is divided into the rate served directly from the cache (cache hit) and the rate that requires uploading from the origin (cache miss). The maximum download rate of the HP Proliant cache server under consideration equals 18 Gb/s. Therefore, during prime time the 1-cache CDN cannot serve all HTTP requests. The shaded area at the top of the figure between the download rate of the 1-cache CDN and the 2-cache CDN represents the requested data volume that is not delivered by the 1-cache CDN. For the 2-cache CDN, the upload rate is higher than for the 1-cache CDN because a 2-cache CDN exhibits less file sharing than a 1-cache CDN.

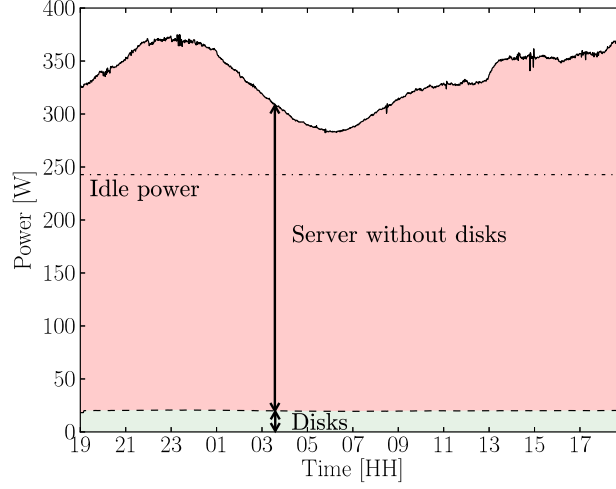


Fig. 6: Single cache-server power consumption over a single day divided between disks and server

Figure 7 shows the power consumed by the 2-cache and 1-cache CDN. Obviously, the 1-cache CDN consumes less power than the CDN composed of 2 cache servers. The total energy consumed over the day under consideration by the 2-cache CDN amounts to 15.25 kWh, whereas the 1-cache CDN consumes only 9.67 kWh. Thus, both from an energy as well as a performance perspective, the 1-cache CDN outperforms the 2-cache CDN during the time periods where the requested download rate does not exceed the maximum rate a single cache server can deliver.

Comparing Figure 7 and 8 reveals that the CDN is far from energy proportional. For example, while the data rate delivered by the CDN decreases from ~ 25 Gb/s to ~ 7.5 Gb/s, i.e., by $\sim 70\%$, the power consumed by the CDN decreases by only $\sim 25\%$. This energy disproportionality is caused by the significant idle power consumed by both cache servers and disks. The CDN is provisioned to cope with the peak load and therefore overprovisioned under moderate to light load. Energy can be saved by the application of *dynamic power management* (DPM), which reduces power consumption by turning off system components or decreasing their performance when they are idle or underutilized [10]. In data-center storage systems, DPM may be enabled by diverting disk accesses from redundant to original disks using a technique called DIV [7]. Similarly we propose to turn off cache servers and disks under moderate to light load. Cache servers and their disks can be powered-down relatively easily because they only contain replicated data, but performance constraints in a real CDN would also need to be considered. When a cache server is powered down, clients can be directed to one of the caches that remain active. When a disk is turned off and the requested file is stored on that disk, the cache server can fetch the file either

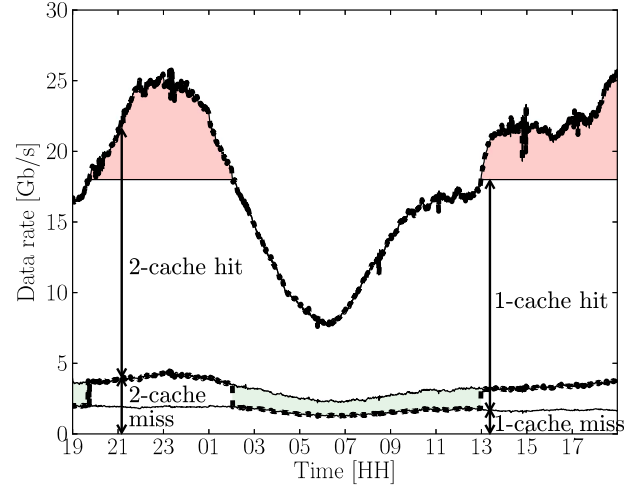


Fig. 7: Download and upload rate (cache miss) for a 1-cache CDN (thin solid), 2-cache CDN (thin solid), and DIV-enabled 2-cache CDN (thick dashed) over a single day

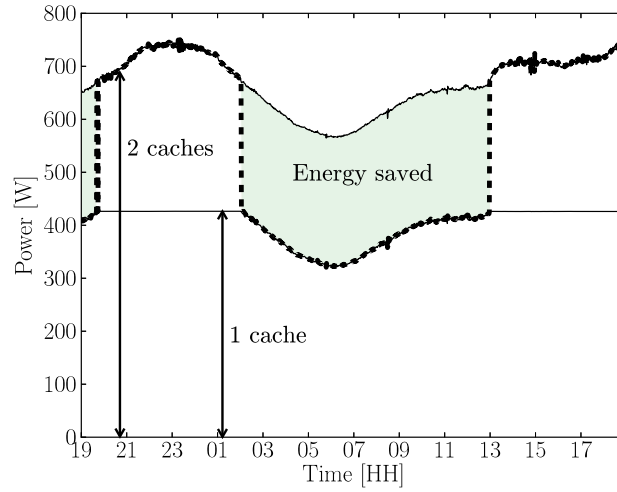


Fig. 8: Power consumed by a 1-cache CDN (thin solid), 2-cache CDN (thin solid), and DIV-enabled 2-cache CDN (thick dashed) over a single day

from one of its neighbors (in the same rack) or from the origin. Devices may be powered down completely or alternatively transitioned to a standby mode. In the future, we intend to adapt DIV to content distribution networks and evaluate this energy-saving technique using our CDN energy simulator.

Figure 8 illustrates DIV-based energy savings. Suppose we turn off one of the caches of the 2-cache CDN whenever the requested download rate is smaller than 18 Gb/s, then we would save 2.84 kWh or, equivalently, 20%, which the shaded areas of Figure 8 represent. The thick dashed line corresponds to the power consumed by the DIV-enabled 2-cache CDN. Even though we provisioned just the minimum number of caches required to handle the workload, the energy savings are significant. Taking a realistic overprovisioning into account (to cope with larger fluctuations caused by, for example, flash crowds), would yield even more energy savings. Additionally, scalability is maintained and bandwidth efficiency even improved as can be observed from Figure 7. The thick dashed lines in Figure 7 represent the download and upload rate of the DIV-enabled 2-cache CDN. The shaded area on the bottom of the figure between the upload rate of the 1-cache and 2-cache CDN represents the additional data volume that can be served directly from the cache.

5 Related Work

Over the last ten years many power-reduction techniques for data-center storage systems were proposed. We present an exhaustive survey of such energy-saving techniques in [4] based on an analysis of over a hundred high-quality papers. To our knowledge, the state-of-the-art in power-aware storage systems does not yet include content distribution networks although such networks are composed of storage elements distributed over multiple data centers. In this paper, we propose to adapt a power-reduction technique called DIV [7] (short for diverted accesses) to content distribution networks as explained in Section 4. The inherent segregation of original and redundant data in a CDN facilitates the application of DIV to CDNs. Instead of diverting accesses from redundant disks, accesses in a CDN are diverted from underutilized cache servers.

Even though power-reduction techniques for storage systems have not yet been applied to content distribution networks, recently the energy efficiency of CDNs in general has become an area of active research. The focus of this research is on energy-aware cache-server placement and file replica placement. [11] compares the energy consumed by delivering content (1) directly from storage servers centralized in a data center, (2) from cache servers of a CDN, and (3) from neighboring clients using P2P delivery. The CDN architecture turns out to be the most energy efficient. Similar results were obtained by [12] in the context of IPTV delivery. [13] claims even higher energy savings by placing the caches in the home gateways. Content-centric networking (CCN) integrates cache servers into routers and therefore eliminates the need for an overlay content distribution network. [14] claims that CCN is even more energy-efficient than a traditional CDN. [15] refines this claim by revealing the dependency on content popularity and catalog size. In this paper, we consider only the CDN architecture and don't take the location of the cache servers into account. Therefore, we don't model the transport energy but only the energy required by the cache servers. However, this server energy is modeled in much greater detail. We use this detailed

cache-server energy consumption model as the basis for a trace-driven CDN energy simulator to evaluate energy-saving techniques based on dynamic power management.

Recently, research results were presented about energy-aware load balancing in content distribution networks [2], which is similar to the energy-saving technique of diverted accesses. However, Mathew et al. focus on the energy-aware load-balancing algorithms, whereas our focus is on a trace-driven CDN energy simulator that allows evaluating potentially different energy-saving techniques. To the best of our knowledge, no CDN energy simulators exist; only CDN performance simulators are available [16].

6 Conclusion

The massive end-user demand for IPTV on any screen spurs the deployment of telco CDNs for HTTP-based adaptive video streaming. The disk-packed cache servers of which such CDNs are composed add to the energy consumption of data centers, which is a growing concern to data-center operators. In this paper, we target energy savings in these content distribution networks. Therefore, we analyze the HTTP-adaptive-streaming workload from an operational CDN delivering IPTV. This workload exhibits cyclic fluctuations, which can be exploited to save energy. In addition, we present a CDN energy simulator, which can be used to evaluate energy-saving techniques in content distribution networks. Simulation results reveal that typical CDNs are far from energy proportional. Therefore, we propose to apply a power-reduction technique for storage systems called DIV to content distribution networks. The adapted technique would turn off complete cache servers or individual cache disks under moderate or light load. In the future, we plan to propose algorithms for determining the minimum number of cache servers and disks that need to stay on to satisfy performance requirements. In addition, we want to make the current simulator location-aware such that we can observe latency and transport energy trade-offs and support geographical load balancing.

Acknowledgments. The authors would like to thank their colleagues of Velocix, an Alcatel-Lucent company, and Koen Laevens for their support in getting access to CDN workload traces. In addition, this work is supported by the Flanders Agency for Innovation by Science and Technology (IWT), grant IWT 100690.

References

1. Alcatel-Lucent: Velocix (2012) <http://www.velocix.com>.
2. Mathew, V., Sitaraman, R., Shenoy, P.: Energy-aware load balancing in content delivery networks. In: Proceedings of the 31st annual IEEE international conference on computer communications. INFOCOM '12, Los Alamitos, CA, USA, IEEE (2012) 954–962

3. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. *Computer* **40** (December 2007) 33–37
4. Bostoen, T., Mullender, S., Berbers, Y.: Power-reduction techniques for data-center storage systems. *ACM Comput. Surv.* **45**(3) (June 2013) 33:1–33:38
5. Cha, M., Rodriguez, P., Crowcroft, J., Moon, S., Amatriain, X.: Watching television over an ip network. In: *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. IMC '08, New York, NY, USA, ACM (2008) 71–84
6. Gill, P., Arlitt, M., Li, Z., Mahanti, A.: Youtube traffic characterization: a view from the edge. In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. IMC '07, New York, NY, USA, ACM (2007) 15–28
7. Pinheiro, E., Bianchini, R., Dubnicki, C.: Exploiting redundancy to conserve energy in storage systems. *SIGMETRICS Perform. Eval. Rev.* **34** (June 2006) 15–26
8. Begen, A., Akgul, T., Baugher, M.: Watching video over the web: Part 1: Streaming protocols. *Internet Computing, IEEE* **15**(2) (March-April 2011) 54–63
9. HP: Hp solid state drive (ssd) for proliant server (2011) http://h18000.www1.hp.com/products/quickspecs/13415_div/13415_div.pdf.
10. Benini, L., Micheli, G.d.: System-level power optimization: techniques and tools. *ACM Trans. Des. Autom. Electron. Syst.* **5** (April 2000) 115–192
11. Feldmann, A., Gladisch, A., Kind, M., Lange, C., Smaragdakis, G., Westphal, F.J.: Energy trade-offs among content delivery architectures. In: *Proceedings of the 9th conference on Telecommunications, Media and Internet Techno-Economics*. CTTE '10, Los Alamitos, CA, USA, IEEE (2010) 1–6
12. Baliga, J., Ayre, R., Hinton, K., Tucker, R.: Architectures for energy-efficient iptv networks. In: *Proceeding of the 2009 Conference on Optical Fiber Communication*. OFC '09, Washington, DC, USA, OSA (2009) 1–3
13. Valancius, V., Laoutaris, N., Massoulié, L., Diot, C., Rodriguez, P.: Greening the internet with nano data centers. In: *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. CoNEXT '09, New York, NY, USA, ACM (2009) 37–48
14. Lee, U., Rimac, I., Kilper, D., Hilt, V.: Toward energy-efficient content dissemination. *IEEE Network* **25**(2) (March 2011) 14–19
15. Guan, K., Atkinson, G., Kilper, D., Gulsen, E.: On the energy efficiency of content delivery architectures. In: *Proceedings of the 4th international workshop on green communications*. GreenComm4, Los Alamitos, CA, USA, IEEE (2011) 1–6
16. Stamos, K., Pallis, G., Vakali, A., Katsaros, D., Sidiropoulos, A., Manolopoulos, Y.: Cdnsim: A simulation tool for content distribution networks. *ACM Trans. Model. Comput. Simul.* **20**(2) (May 2010) 10:1–10:40